An Automatic Trust Calculation Based on the Improved Kalman Filter Detection Algorithm

Bo Ye, Mohammad Ghavami, Anjum Pervez, Maziar Nekovee

An Automatic Trust Calculation Based on the Improved Kalman Filter Detection Algorithm

Bo Ye[1], Mohammad Ghavami[1], Anjum Pervez[1], and Maziar Nekovee[2]

[1] ESBE, London South Bank University, London SE1 0AA, UK
{yeb, ghavamim, perveza}@lsbu.ac.uk
[2] BT Research, Martlesham, Suffolk IP5 3RE, UK
maziar.nekovee@bt.com

Abstract. In service-oriented systems, how much a service can be trusted is increasingly crucial for service consumers to make the best decision. Because the methods for deriving trust based on manually assigned feedback cost much time and suffer several drawbacks, automatic trust calculation is the only feasible method for large-scale service-oriented applications. Therefore an automatic trust calculation using other non-trust quality criterion values is proposed. To make the calculation accurate, the Kalman Filter is employed to filter out malicious non-trust values instead of directly filtering out malicious trust values. Furthermore, an improved algorithm is proposed by taking the relationship between the non-trust criterion value and its variance into account to offer higher detection accuracy. Although malicious data can be filtered out, dishonest or inaccurate values can still influence trust values. Hence similarity between consumers is used to weight the values from others. Finally, experiments are carried out to access the validation and robustness of our model. The results show that our algorithm can offer higher detection accuracy under more strategic malicious situations.

Keywords: Web service; Quality of Service; Trust; Kalman Filter

1 Introduction

The rapid growth of service-oriented applications has spurred a considerable amount of research in this field. However, service providers are usually little known by service consumers. Among a great number of service providers providing identical or similar services with varying Quality of Service (QoS), it is hard for service consumers to select appropriate services. To help consumers make decisions, QoS of service-oriented systems has been modelled [1–3, 10] and various selection algorithms have been proposed to optimize the results of service selection. A number of QoS properties have been used to evaluate services in these models. These novel centralized systems [1–3, 10] modelled QoS properties of services, and then based on consumer requirements and the values of QoS published by service providers, many different optimization algorithms were employed to select appropriate services for service composition. However, they did not focus

on how these values of QoS properties were collected. If all these values are given by service providers, they may not be true, due to dishonesty of certain service providers. Hence, among these QoS criteria, trust is increasingly important for service consumers to select service providers. In addition the systems [2, 10] did not aggregate the values of trust, while in the approaches [1, 3] trust is just one of the QoS criteria considered.

Furthermore, some systems [13] also collect feedback from service consumers to evaluate service providers. Similarly service consumers may also provide malicious feedback. Although these approaches were effective to select services for service consumers, they did not focus on the detection of malicious service consumers. Therefore if malicious service consumers exist, they may not select appropriate services for service consumers.

Several trust-based systems [5, 14, 15, 21] have been proposed, however the authors in [5] did not focus on detecting malicious consumers. Although they provided high flexibility for consumers to use a variety of scoring functions over the same data for personalized reputation evaluation, this approach is based on the assumption that consumers do not mask their malicious behaviour, meaning that it is hard to detect those malicious service consumers that behave well until they gain good trust values and then behave maliciously. The approaches in [14, 15, 21] identified malicious consumers to minimize their threat and protect the system from possible misuses and abuses by them. The authors in [15] represented trust by the use of two attributes, trust value and trust estimation variance, and employed the Kalman Filter to filter out and to aggregate feedback. Their experiments showed that the model provided higher robustness to estimate trust values with a lower false detection rate. In [14], the authors adopted the method of cumulative sum to detect malicious feedback for a reputation-based system. In [21], the authors required service consumers to provide a trust value on each QoS property of a service, meaning that a value of trust in one QoS property indicates how much this QoS property of this service can be trusted. Meanwhile, the authors in [14] required one trust value for one service each time a service was invoked. Both [14] and [21] employed Euclidean distance to calculate similarity between two service consumers, which was used as the weight to aggregate the data collected from other consumers.

Although these approaches [14, 15, 21] are effective, the authors in [15] did not focus on how reputation values were collected while the authors in [14, 21] required humans to provide feedback ratings used as trust values. It would be better to calculate the value of reputation automatically, because feedback ratings provided by humans are not accurate, due to differences in humans' preference and incomplete knowledge. In addition, collecting feedback from humans becomes harder in large-scale service-oriented applications. The authors in [12] surveyed techniques of trust in service workflows and relevant contexts. According to their review, they also discovered that lacking of a unified way to formalize trust prevented automation in trust-related processes from realization.

Hence, trust is increasingly crucial in service-oriented systems for service consumers to select appropriate services, and trust is also important for other

applications, such as e-commerce environments [7, 16], mobile adhoc networks [4,6] and peer-to-peer networks [8,9,20,22]. The authors in [7] reviewed existing and proposed trust-based commercial online applications. Some trust issues and an overview of trust-based evaluation approaches in e-commerce environments were also given in [16].

In [19,22], a Bayesian reputation approach was proposed to calculate the trust value based on the beta probability density functions (PDF). In [19], intuitive parameters needed to be tuned manually without guarantee of any quantitative confidence. In [17], the trust was modelled as a three-dimension belief (b, d, u), which represented the positive, negative and uncertain probabilities. Although the trust in [17, 19, 22] was modelled as predicted probability values, prediction variance was ignored by them, which was considered in [15]. The authors in [15] proposed a general trust model for a more robust reputation evaluation. However, all these models only use feedback between service providers and consumers, but did not use feedback among service consumers.

Many researchers have proposed various approaches by employing different techniques to evaluate the trust in web services. Although a variety of efficient and robust measure solutions [5, 11, 14, 15, 21] have been proposed, these approaches still mostly have following weaknesses, according to a brief review above.

First of all, most approaches measure service reputation based on the assumption that the feedback are provided by humans, and therefore it is difficult to ensure the accuracy. Furthermore, this kind of system cannot be built without humans participating. Manually assigned feedback cost much time and have several disadvantages. For instance, certain humans are not willing to provide feedback and may provide unfair ones. The method of automatic trust calculation is the only feasible one for large-scale service-oriented applications.

Secondly, existing trust measure approaches mostly collect feedback only between service consumers and providers, and rarely use data among consumers. Most of such systems are centralized, otherwise it is hard for service consumers to obtain the information on trust of service providers. Therefore, how a service consumer can obtain the trust level of a service in distributed systems and how data collected from other service consumers is aggregated need to be considered.

Lastly, due to existing malicious service consumers, how those malicious ones can be filtered out is increasingly important. Malicious service consumers might provide malicious values to falsely improve the trust in certain service providers, or to degrade the trust in certain service providers for commercial benefits.

To address the weaknesses above, an approach to measure the trust both in service providers and consumers has been proposed. This approach first groups service quality criteria, and then measures the trust in each quality criterion of a service based on their characteristics. The measure of trust in service providers has also been divided into two main stages, including Time Domain and Aggregation Domain, because of the following two reasons. First, one service consumer may invoke a service many times, so that it can measure the trust in the service based on the data collected by the consumer itself at Time Domain. At this

stage, all data are obtained by itself, and therefore it is unnecessary to filter out any information. Second, it may also compute the value of trust by the use of data from other service consumers. At this stage, the service consumer not only needs to aggregate all data, but also has to filter out malicious data. This stage is also called Aggregation Domain.

Compared to the existing approaches, our main contributions have been summarized as follows:

First, Quality of Web Service Criteria have been grouped into several classes based on their characteristics, and trust calculation has been divided into two main domains. In addition, most existing approaches treat trust as only one QoS criterion of a service, equivalent to response time, meaning that one service has only one value representing the overall reputation of the service. However we compute a value of trust in each QoS criterion.

Second, the majority of existing approaches directly filter out malicious values of trust provided by other service consumers. Different from them, we filter out malicious values of other non-trust QoS criteria, and then all these QoS criteria values are aggregated to compute the trust values automatically.

Lastly, at Aggregation Domain, when a service consumer aggregates the data from others, the trust in other consumer X is employed to weight the trust value in a service provider, which is provided by that consumer X. The value of trust in other consumer is calculated using distance functions.

The rest of this paper is organized as follows. Section 2 demonstrates how quality criteria are classified and how the values of trust in quality criteria and in consumers are calculated automatically. Section 3 presents how malicious values are detected and how different values from a variety of consumers are aggregated. The model is evaluated by carrying out different experiments in Section 4. The final Section contains the conclusion and some ideas for further work.

2 Quality Criterion

In this section, some concepts related to trust are introduced first. Based on these concepts, this section presents how the value of the trust in each individual criterion and other service consumers are calculated.

Quality Criterion (QC) encompasses a number of QoS properties used to evaluate a web service, including Price (P), Response Time (RT), Availability (A), Success Rate (SR) and Trust (T). There are two ways to categorize QC.

First, based on the way how it affects the overall QoS of a service, QC can be classified as either Positive Criterion, whose increase benefits the overall QoS, or Negative Criterion whose decrease benefits the overall QoS.

Second, on the basis of the nature of a criterion, criteria then fall into two major classes including Ratio QC (RQC) and Non-ratio QC (NRQC). RQC's values can be presented as a ratio, which can be directly used as the trust of the criterion, such as availability, success rate and so on. Please note that RQC are not the criteria whose values obtained from service providers are rate. For example, compensation rate, whose values gotten from providers are rate, is not

a RQC. NRQC's values cannot be presented as a ratio, eg. price, response time etc.

For a criterion of a service, a service consumer can get its value in two different ways. The service consumer can obtain the value of a criterion published by the service provider, which is called as Published Value (PV), or it can get the value by invoking the service, which is named Actual Value (ACV). PV is published by providers, and it can be updated at any time. ACV is collected by consumers, which may be different from PV. For instance, a provider may publish $40ms$ as a service's response time, but the actual response time may be $43ms$ when a consumer invokes the service.

Trust can be seemed as one QC and it is denoted by T. One consumer A has the value of the trust in another one B, meaning that A knows how much he can trust B. Trust can be classified as either criterion or reference trust, on the basis of trust purpose. For an illustrative purpose, a criterion C of a service S provided by a service provider P is denoted by $P.S.C$.

- Criterion Trust: A's trust in $P.S.C$, denoted by $T(A \rightarrow P.S.C)$. It identifies how much the service's criterion $P.S.C$ can be trusted.
- Reference Trust: A's trust in consumer B's capacity of referring to other consumers' ability to do something, defined by $T(A \rightarrow B)$. Please note that a service provider can also have a reference trust, because the service provider can also be a service consumer, recommending another service provider.

Based on the ways how it affect the overall trust, a trust can be divided into Positive Part, which increases the trust, and Negative Part, which decreases it. Similarly, the actual value of a criterion can also be classified as either Positive Actual Value, which increases the trust of the criterion, or Negative Actual Value, which decreases the criterion's trust.

2.1 Criterion Trust Calculation

After demonstrating relevant concepts, how the value of the trust is derived from values of other non-trust quality criteria is presented. To begin with, a few notations are introduced first: $T(A \rightarrow P.S.C)_j$ is the value of A's trust in $P.S.C$ after j^{th} time A invokes a service S; c represents the published value of $P.S.C$; c_j is the actual value obtained by A after j^{th} time invoking service S.

Assume that Criterion C is a negative one, meaning that the decrease of this criterion benefits the trust, and then $T(A \rightarrow P.S.C)_j$ is calculated by the following equations.

Number of positive C values,

$$\mathrm{num}_j^{po} = \begin{cases} \mathrm{num}_{j-1}^{po} + 1 & c_j \leq c \\ \mathrm{num}_{j-1}^{po} & c_j > c \end{cases}$$

Number of negative C values,

$$\mathrm{num}_j^{ne} = \begin{cases} \mathrm{num}_{j-1}^{ne} & c_j \leq c \\ \mathrm{num}_{j-1}^{ne} + 1 & c_j > c \end{cases}$$

The value of positive part of $T(A \to P.S.C)_j$ is calculated by:

$$T_j^{po} = \begin{cases} \sqrt{\frac{\text{num}_{j-1}^{po}(T_{j-1}^{po})^2 + (1-\frac{c_j}{c})^2}{\text{num}_j^{po}}} & c_j \leq c \\ T_{j-1}^{po} & c_j > c \end{cases}$$

Value of negative part of $T(A \to P.S.C)_j$,

$$T_j^{ne} = \begin{cases} T_{j-1}^{ne} & c_j \leq c \\ \sqrt{\frac{\text{num}_{j-1}^{ne}(T_{j-1}^{ne})^2 + (1-\frac{c_j}{c})^2}{\text{num}_j^{ne}}} & c_j > c \end{cases}$$

At last, the value of A's trust in $P.S.C$ is computed as follows:

$$T(A \to P.S.C)_j = 1 + T_{j-1}^{po} - \frac{\text{num}_j^{ne} \cdot T_j^{ne}}{\text{num}_j^{ne} + \text{num}_j^{po}}$$

Please note that the values which are equal to the PV are always classified as the positive ones.

2.2 Reference Trust Calculation

In order to aggregate data from other service consumers, a service consumer needs to know how much he can trust others. In this paper similarity between two consumers is used as a service consumer A's reference trust in another consumer B, because A can trust B more, if values of the trust maintained by A are more similar to B's. Using the value of trust in B, A can know how much he can trust the services or other consumers referred by B.

$$T(A \to B) = 1 - \left(\frac{\sum_P \sum_S \sum_C (T(A \to P.S.C) - T(B \to P.S.C))^p}{|T(A \to P.S.C)|} \right)^{\frac{1}{p}} \quad (1)$$

Equation. (1) is the mostly used. When $p = 2$, this equation is the Euclidean distance function used in [14, 21].

2.3 Trust Transitivity

Based on the transitivity of trust, the method of weighted mean aggregation is used to aggregate values of the trust in a certain service from other service consumers. Similarity between consumers is used as the weight. For instance, a consumer A needs to know how much he can trust in $P.S.C$, however he has no information about it. In addition, B has trust in $P.S.C$, denoted by $T(B \to P.S.C)$, and A knows how much he can trust B, represented by $T(A \to B)$.

Therefore A's trust in $P.S.C$, $T(A \to P.S.C)$, can be derived by aggregating B's function trust in $P.S.C$ and A's trust in B as follows:

$$T(A \to P.S.C) = T(A \to B) \cdot T(B \to P.S.C)$$

It is common to collect reference trusts from several different service users to make better decisions. This can be called consensus trust. Assume a service consumer A needs to obtain the value of the trust in $P.S.C$, and he knows little about $P.S.C$. However, he has information about trust in other consumers X and Y, and both of they have a trust in $P.S.C$, denoted by $T(X \to P.S.C)$ and $T(Y \to P.S.C)$ respectively. The consensus trust of X and Y's trust in $P.S.C$ is a trust that reflects trust in a fair and equal way, and derived by:

$$T(A \to P.S.C) = \frac{|T(A \to X) \cdot T(X \to P.S.C)| + |T(A \to Y) \cdot T(Y \to P.S.C)|}{|T(A \to X)| + |T(A \to Y)|}$$

3 Criterion Value Estimation and Improved Malicious Value Detection

Malicious service consumer can be classified as either adulating service consumer, which tries to falsely improve the trust in certain service providers, or defaming service consumer, trying to degrade the trust in certain service providers. The algorithm of malicious value detection in [15] has been shown that it is better than Bayesian-based, and many other algorithms. Hence, Kalman Filter-based algorithm is adopted to detect malicious values. The authors in [15] used this algorithm to filter out malicious values of the trust, however we use it to filter out malicious values of non-trust quality criteria to retain the accuracy of the trust. In addition, based on the algorithm we not only estimate the value of non-trust quality criteria, but also predict its variance. We further improve this algorithm by taking the relationship between the value of non-trust quality criteria and its variance into account.

3.1 Criterion Value Estimation

Because the value of a criterion C of a service S provided by a service provider P obtained by a service consumer A each time is independent, it is reasonable to model the distribution of the values of $P.S.C$ as Normal distribution. For each criterion, its values follow normal distribution with $\{\mu^r, \sigma^r\}$, where μ^r is the real value of $P.S.C$'s μ, and σ^r is the actual $P.S.C$'s variance.

Assume that A is going to use estimated values from other consumers to predict $P.S.C$'s $\{\mu^r, \sigma^r\}$. Service consumer i's estimated values of $P.S.C$'s $\{\mu^r, \sigma^r\}$ are denoted as $\{\mu^e_i, \sigma^e_i\}$. After aggregating i's estimated values, A's estimated values are denoted as $\{\mu^e_{A,i}, \sigma^e_{A,i}\}$. Because of incomplete knowledge of $P.S.C$, i's estimated values usually have a deviation from A's estimated values $\{\mu^e_{A,i}, \sigma^e_{A,i}\}$.

Because the estimated values are from independent consumers, the relation between i's estimate and A's estimate is modelled as follows:

$$\begin{aligned} \mu_i^e &= \mu_{A,i}^e + \lambda_\mu \text{ and } p(\lambda_\mu) \sim \text{Normal}(0, \Lambda_\mu) \\ \sigma_i^e &= \sigma_{A,i}^e + \lambda_\sigma \text{ and } p(\lambda_\sigma) \sim \text{Normal}(0, \Lambda_\sigma) \end{aligned} \tag{2}$$

Note that λ_μ is different from σ_i^e. λ_μ is an estimate noise covariance when A estimates the real value μ^r, while σ_i^e is estimated covariance from service consumer i, which may be malicious. Similarly, λ_σ is an estimate noise covariance when A estimates the real value σ^r.

Based on the Kalman Filter [18], the estimation of $\{\mu^r, \sigma^r\}$ is governed by the linear stochastic difference equations:

$$\begin{aligned} \mu_{A,i}^e &= F_\mu \mu_{A,i-1}^e + Bu_{i-1} + w_{\mu,i-1};\ p(w_\mu) \sim \text{Normal}(0, W_\mu) \\ \sigma_{A,i}^e &= F_\sigma \sigma_{A,i-1}^e + Bu_{i-1} + w_{\sigma,i-1};\ p(w_\sigma) \sim \text{Normal}(0, W_\sigma) \end{aligned}$$

where, F is the factor for relationship between the previous estimate based on the estimate of service consumer $i-1$ and the current estimate based on i's estimate, and u is the optional control input to the estimate $\{\mu_A^e, \sigma_A^e\}$. Because in our model there is no control input, u is 0. Hence, our estimates are governed by the following linear difference equations:

$$\begin{aligned} \mu_{A,i}^e &= F_\mu \mu_{A,i-1}^e + w_{\mu,i-1};\ p(w_\mu) \sim \text{Normal}(0, W_\mu) \\ \sigma_{A,i}^e &= F_\sigma \sigma_{A,i-1}^e + w_{\sigma,i-1};\ p(w_\sigma) \sim \text{Normal}(0, W_\sigma) \end{aligned}$$

In the Kalman Filter, there are two steps: *Predict* step and *Update* step. P_μ and P_σ represent predict error covariance of $\mu_{A,i}^e$ and $\sigma_{A,i}^e$ respectively. The *Predict* step is responsible for obtaining the priori estimate, denoted by $\{\bar{\mu}_{A,i}^e, \bar{\sigma}_{A,i}^e\}$, for the next step based on the previous estimate $\{\mu_{A,i-1}^e, \sigma_{A,i-1}^e\}$. Similarly, priori predict error covariances are denoted by $\bar{P}_\mu$ and $\bar{P}_\sigma$. The *Update* step is responsible for incorporating a new service consumer's estimate $\{\mu_i^e, \sigma_i^e\}$ to obtain an improved posteriori estimate $\{\mu_{A,i}^e, \sigma_{A,i}^e\}$.

Predict step:

$$\bar{\mu}_{A,i}^e = F_{\mu,i} \mu_{A,i-1}^e, \quad \bar{\sigma}_{A,i}^e = F_{\sigma,i} \sigma_{A,i-1}^e \tag{3}$$

$$\bar{P}_{\mu,i} = F_{\mu,i}^2 P_{\mu,i-1} + W_{\mu,i}, \quad \bar{P}_{\sigma,i} = F_{\sigma,i}^2 P_{\sigma,i-1} + W_{\sigma,i} \tag{4}$$

Update step:

$$K_{\mu,i} = \frac{\bar{P}_{\mu,i}}{\bar{P}_{\mu,i} + \Lambda_{\mu,i}}, \quad K_{\sigma,i} = \frac{\bar{P}_{\sigma,i}}{\bar{P}_{\sigma,i} + \Lambda_{\sigma,i}} \tag{5}$$

$$\begin{aligned} \mu_{A,i}^e &= \bar{P}_{\mu,i} + K_{\mu,i}(\mu_i^e - \bar{\mu}_{A,i}^e), \\ \sigma_{A,i}^e &= \bar{P}_{\sigma,i} + K_{\sigma,i}(\sigma_i^e - \bar{\sigma}_{A,i}^e) \end{aligned} \tag{6}$$

$$P_{\mu,i} = (1 - K_{\mu,i})\bar{P}_{\mu,i}, \quad P_{\sigma,i} = (1 - K_{\sigma,i})\bar{P}_{\sigma,i} \tag{7}$$

In order to compute the parameters $F_{\mu,i}$, $\Lambda_{\mu,i}$, $W_{\mu,i}$, $F_{\sigma,i}$, $\Lambda_{\sigma,i}$, $W_{\sigma,i}$, the following equations are used:

$$F_{\mu,i} = \frac{\sum_{j=1}^{i-1} \mu_{A,j}^e \mu_{A,j-1}^e}{\sum_{j=1}^{i-1} (\mu_{A,j}^e)^2}, \quad F_{\sigma,i} = \frac{\sum_{j=1}^{i-1} \sigma_{A,j}^e \sigma_{A,j-1}^e}{\sum_{j=1}^{i-1} (\sigma_{A,j}^e)^2} \tag{8}$$

$$\Lambda_{\mu,i} = \frac{1}{i} \sum_{j=1}^{i-1} (\mu_j^e - \mu_{A,j}^e)^2, \quad \Lambda_{\sigma,i} = \frac{1}{i} \sum_{j=1}^{i-1} (\sigma_j^e - \sigma_{A,j}^e)^2 \tag{9}$$

$$\begin{aligned} W_{\mu,i} &= \frac{1}{i} \sum_{j=1}^{i-1} (\mu_{A,j}^e - F_i \mu_{A,j-1}^e)^2, \\ W_{\sigma,i} &= \frac{1}{i} \sum_{j=1}^{i-1} (\sigma_{A,j}^e - F_i \sigma_{A,j-1}^e)^2 \end{aligned} \tag{10}$$

3.2 Improved Malicious Value Detection

Given significance probability levels δ_μ and δ_σ, the problem of determining if a consumer i is not malicious is to find the threshold values $\Delta_{\mu,i}$ and $\Delta_{\sigma,i}$ so that:

$$P(|\mu_i^e - \mu_{A,i}^e| \leq \Delta_{\mu,i}) = \delta_\mu, P(|\sigma_i^e - \sigma_{A,i}^e| \leq \Delta_{\sigma,i}) = \delta_\sigma \tag{11}$$

In addition, $\mu_i^e - \mu_{A,i}^e$ and $\sigma_i^e - \sigma_{A,i}^e$ follow zero mean normal distribution with variance $P_{\mu,i} + \Lambda_{\mu,i}$ and $P_{\sigma,i} + \Lambda_{\sigma,i}$ respectively. Hence, there are also equations:

$$\begin{aligned} P(|\mu_i^e - \mu_{A,i}^e| \leq \Delta_{\mu,i}) &= 1 - 2\Phi\left(\frac{-\Delta_{\mu,i}}{\sqrt{P_{\mu,i} + \Lambda_{\mu,i}}}\right), \\ P(|\sigma_i^e - \sigma_{A,i}^e| \leq \Delta_{\sigma,i}) &= 1 - 2\Phi\left(\frac{-\Delta_{\sigma,i}}{\sqrt{P_{\sigma,i} + \Lambda_{\sigma,i}}}\right) \end{aligned} \tag{12}$$

where $\Phi(x)$ is the cumulative distribution function of the standard normal distribution. Hence, after solving Equation. (11) and (12), $\Delta_{\mu,i}$ and $\Delta_{\sigma,i}$ can be obtained:

$$\begin{aligned} \Delta_{\mu,i} &= -\Phi^{-1}\left(\frac{1-\delta_\mu}{2}\right)\sqrt{P_{\mu,i} + \Lambda_{\mu,i}}, \\ \Delta_{\sigma,i} &= -\Phi^{-1}\left(\frac{1-\delta_\sigma}{2}\right)\sqrt{P_{\sigma,i} + \Lambda_{\sigma,i}} \end{aligned} \tag{13}$$

Using the threshold values $\Delta_{\mu,i}$ and $\Delta_{\sigma,i}$, malicious values of μ^e and σ^e can be detected respectively. However, a malicious consumer can still manipulate the model by setting σ^e or μ^e to be the lower or upper limit. Although a malicious consumer i can set its feedback $\{\mu_i^e, \sigma_i^e\}$ to be upper or lower limit, the probability of such feedback may be very low or even zero. Hence, in order to

improve the accuracy, Mahalanobis equation is adopted, which is a measurement between two vectors. After Kalman Filter-based estimation and roughly filtering out using thresholds, a two-column vector is obtained, with the two columns representing estimate real value and estimate real variance respectively, denoted by $\overrightarrow{x} = \{\overrightarrow{\mu^e}, \overrightarrow{\sigma^e}\}$. Each row represents one estimate at each step based on the estimates provided a certain service consumer. The Mahalanobis values between i^{th} row $\overrightarrow{x_i}$ and $\overrightarrow{x}$ are calculated by the following equation, and the Mahalanobis vector is denoted by M.

$$m_i = \sqrt{(E(\overrightarrow{x_i}) - E(\overrightarrow{x}))^T S^{-1} (E(\overrightarrow{x_i}) - E(\overrightarrow{x}))} \tag{14}$$

(where S represents the pooled covariance matrix calculated with $\overrightarrow{x_i}$ and $\overrightarrow{x}$)

Given significance probability levels δ_m, the problem of determining if the i^{th} row is not malicious is to find the threshold values Δ_m so that:

$$P(m_i \leq \Delta_m) = \delta_m \tag{15}$$

In addition, M follows Gaussian distribution with mean μ_m and variance σ_m. Hence, there are also equations:

$$P(m_i \leq \Delta_m) = \Phi(\frac{\Delta_m - \mu_m}{\sqrt{\sigma_\mu}}) \tag{16}$$

where $\Phi(x)$ is the cumulative distribution function of the standard normal distribution. Hence, after solving Equation. 15 and 16, Δ_m can be obtained:

$$\Delta_m = \Phi^{-1}(\delta_m)\sqrt{\sigma_m} + \mu_m \tag{17}$$

3.3 Calculation Algorithm

Each time a service consumer invokes a service, it can obtain values of all criteria. Then this service uses these values, which are called values of quality criterion at Time Domain, used to estimate real criterion values. Because at time domain all values are collected by service consumer itself, it is unnecessary to filter out malicious values. For RQC, it is easy to calculate the values and it is accurate. For instance, the success rate c_{sr} of a service S can be calculated by the following equations:
If the i^{th} invocation of S is successful:

$$\text{num}_{\text{success},i} = \text{num}_{\text{success},i-1} + 1$$

If the i^{th} invocation of S fails:

$$\text{num}_{\text{success},i} = \text{num}_{\text{success},i-1}$$

Finally:

$$\text{num}_{\text{total},i} = \text{num}_{\text{total},i-1} + 1; \quad c_{sr} = \frac{\text{num}_{\text{success},i}}{\text{num}_{\text{total},i}}$$

Each time the service S invoked by a service consumer j, it can get values of NRQC. If exact values C_j of NRQC price C can be obtained, the value $\{\mu_j^e, \sigma_j^e\}$ can also be calculated as follows:

$$\mu_j^e = E[C_j]; \sigma_j^e = E[(C_j - \mu_j^e)^2]$$

However, exact values of certain NRQC cannot be obtained, such as response time. Not only because computer is a complex dynamic system, but also because of network delay, it is impossible to get exact response time of a service. Hence, at this point the method of criterion value estimation in Section 3 is used to estimate the value $\{\mu_j^e, \sigma_j^e\}$.

Each service consumer can collect criterion values of various services from numerous service consumers, and these values can be aggregated to estimate real criterion values, although certain values may be malicious. These values are called values of quality criterion at Aggregation Domain. At aggregation domain, values of all criteria including RQC and NRQC are estimated by using the method of criterion value estimation in Section 3, not only because malicious values need to be filtered out, but also because all these values may not be accurate due to incomplete knowledge on service providers.

Assume a service consumer A is going to aggregate values from other consumers to calculate the trust in $P.S.C$. Each step after estimating the value of $P.S.C$ by the use of the value provided by a consumer X, the trust in $P.S.C$ is calculated using the method presented in Section 2.1. A's reference trust in X is calculated by using the method introduced in Section 2.2, which is used to weight the trust value in $P.S.C$ from X. Furthermore, second hand values are aggregated with the approach presented in Section 2.3.

4 Performance Evaluation

In this section, our trust model is evaluated in a simulated environment. Experiments are carried out to evaluate the robustness of this model, compared to the algorithm in [15]. Because the authors in [15] have shown that it is better than Bayesian-based algorithms and so on, our model is not going to be compared to those approaches again.

In order to evaluate the robustness of this model, experiments are carried out in an environment with malicious values. Our approach is compared with the method in [15], represented by RLM for illustration purpose, to evaluate the robustness of malicious value detection, and IKM is short for our model, Improved Kalman Filter. The probability of malicious values is set up to 40%, because it is almost impossible that more than a half of service consumers behave maliciously and it is impossible to perform well in an environment with more than 50% malicious service consumers.

True Malicious Rate is the percentage of correctly detected malicious values. The number of all malicious values and correctly detected malicious ones are denoted by num_m and num_c respectively, and then true malicious rate is calculated by $\dfrac{\text{num}_c}{\text{num}_m}$.

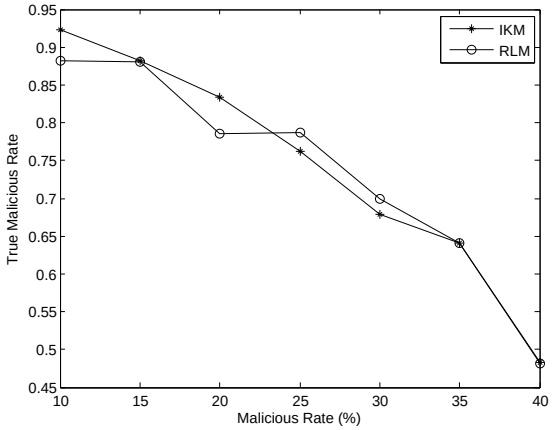

Fig. 1. Average true malicious rate with variance being set to be an extreme low value

False Malicious Rate is the percentage of wrongly detected non-malicious values. The number of all non-malicious and wrongly detected malicious values are denoted by num_{non} and num_w respectively, and then false malicious rate is calculated by $\frac{\text{num}_w}{\text{num}_{\text{non}}}$.

In this experiment, similar to the experiment in [15], the variance σ^e of malicious values is set to be an extreme low value. All results are shown in Fig. 1 and 2, and our model performs almost the same as the original RLM algorithm in this situation, as the increase of the malicious value probability. Because our algorithm is based on RLM, it does not perform worse than RLM. IKM aims at another situation which is not considered by RLM. Hence, in this experiment, IKM does not outperform RLM, either. Results in Fig. 1 and 2 further proved these two points.

Furthermore, in another experiment three following types of malicious values are added:

- σ^e is set to be the real value and μ^e is set to be the lower or upper limit;
- σ^e is set to be the lower or upper limit and μ^e is set to be real value;
- Both σ^e and μ^e are set to be the lower or upper limit.

The values of σ^e in our model corresponds to P in two dimension tuple, $\{\langle R\rangle, P\}$, of RLM. RLM used manual feedback and the values of P were also set manually by consumers. Although in our model both σ^e and μ^e are derived from the collected values, consumers can still manipulate the values and provided inaccurate values to other consumers. In this situation, malicious consumers try to mislead the results gradually. While there is no relationship between R and P in RLM, in IKM σ^e and μ^e are related to each other. Hence, seen from Fig. 3, our model can still have high detection efficiency in this strategic malicious environment while RLM does not. However, as shown in Fig. 4, our model suffers higher false detection rate than RLM.

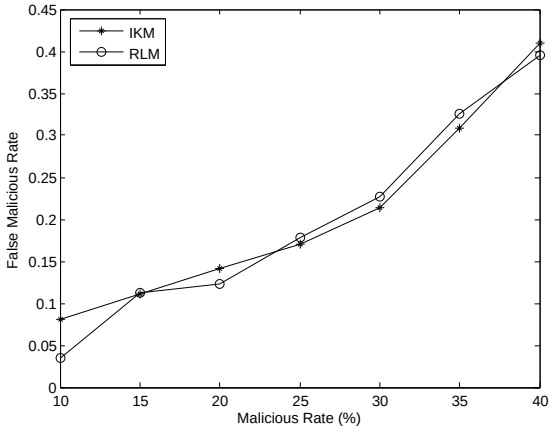

Fig. 2. Average false malicious rate with variance being set to be an extreme low value

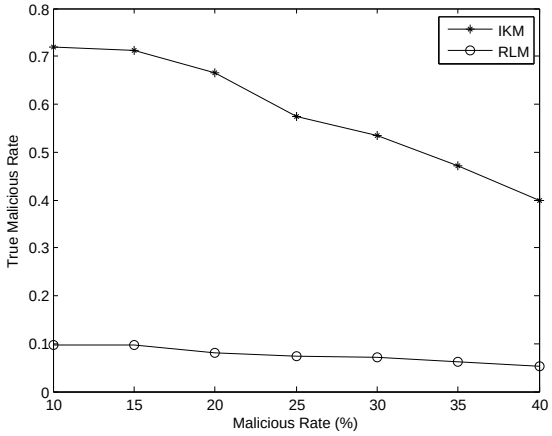

Fig. 3. Average true malicious rate with three types of malicious values

5 Conclusion

Trust in QoS of service providers is increasingly important for service consumers to select appropriate services. In this paper, QoS criteria have been classified into several groups on the basis of their characteristics. Based on the trust in service providers and other consumers, an automatic algorithm of trust calculation has been presented. This model significantly helped reduce the influence of dishonest service consumers. The trust calculation process has been divided into two steps, Time and Aggregation Domain. At time domain, a service consumer used the values obtained by the consumer itself while at aggregation domain a service consumer calculated the value of trust using values from others, which may be malicious. Hence at aggregation domain, an improved algorithm based on the

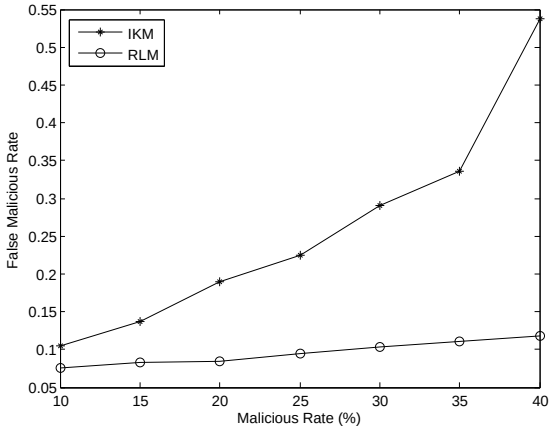

Fig. 4. Average false malicious rate with three types of malicious values

Kalman Filter has been presented to filter out malicious values and the trust in other consumer X was used to weight the trust provided by X. Finally, our model has been evaluated by several experiments and the results have shown that a more accurate value estimation can be made by our model, with higher detection accuracy than the original algorithm under a more strategic malicious environment, although our algorithm suffers a higher false detection rate.

However, our model required a large amount of historic estimation and lots of calculation. Hence, further research will be carried out to reduce the need of storing a large number of historic estimation and calculation. In addition, when aggregating values from other consumers, only first-hand data were used, meaning that a service consumer only aggregates the data from the service consumers that it knows. In future, more data will be considered.

Acknowledgment

This work is partially supported by British Telecommunications Research. The authors would like to thank Keith Briggs, Michael Fitch, Santosh Kawade etc.

References

1. Ardagna, D., Pernici, B.: Adaptive service composition in flexible processes. Software Engineering, IEEE Transactions on 33(6), 369–384 (Jun 2007)
2. Calinescu, R., Grunske, L., Kwiatkowska, M., Mirandola, R., Tamburrelli, G.: Dynamic QoS management and optimization in service-based systems. Software Engineering, IEEE Transactions on 37(3), 387–409 (May-Jun 2011)
3. Cardellini, V., Casalicchio, E., Grassi, V., Iannucci, S., Lo Presti, F., Mirandola, R.: MOSES: A Framework for QoS Driven Runtime Adaptation of Service-Oriented Systems. Software Engineering, IEEE Transactions on 38(5), 1138–1159 (Sep-Oct 2012)

4. Cho, J.H., Swami, A., Chen, I.R.: A survey on trust management for mobile ad hoc networks. Communications Surveys Tutorials, IEEE 13(4), 562–583 (2011)
5. Conner, W., Rouvellou, I., Iyengar, A., Nahrstedt, K., Mikalsen, T.: A trust management framework for service-oriented environments. In: International World Wide Web Conference, 2009. pp. 891–900 (2009)
6. Govindan, K., Mohapatra, P.: Trust computations and trust dynamics in mobile adhoc networks: A survey. Communications Surveys Tutorials, IEEE 14(2), 279 –298 (2012)
7. Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. Decision Support Systems 43(2), 618–644 (Mar 2007)
8. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The eigentrust algorithm for reputation management in p2p networks. In: Proceedings of the 12th international conference on World Wide Web. pp. 640–651. WWW '03, ACM, New York, NY, USA (2003)
9. Li, X., Zhou, F., Yang, X.: Scalable Feedback Aggregating (SFA) Overlay for Large-Scale P2P Trust Management. Parallel and Distributed Systems, IEEE Transactions on 23(10), 1944–1957 (Oct 2012)
10. Moser, O., Rosenberg, F., Dustdar, S.: Domain-specific service selection for composite services. Software Engineering, IEEE Transactions on 38(4), 828–843 (Jul-Aug 2012)
11. Nepal, S., Malik, Z., Bouguettaya, A.: Reputation Management for Composite Services in Service-Oriented Systems. International Journal of Web Services Research 8(2), 29–52 (2011)
12. Viriyasitavat, W., Martin, A.: A survey of trust in workflows and relevant contexts. Communications Surveys Tutorials, IEEE 14(3), 911 –940 (2012)
13. Wang, S., Sun, Q., Yang, F.: Quality of service measure approach of web service for service selection. IET Software 6(2), 148–154 (Apr 2012)
14. Wang, S., Sun, Q., Zou, H., Yang, F.: Reputation measure approach of web service for service selection. IET Software 5(5), 466–473 (Oct 2011)
15. Wang, X.F., Liu, L., Su, J.S.: RLM: A general model for trust representation and aggregation. Services Computing, IEEE Transactions on 5(1), 131–143 (Jan-Mar 2012)
16. Wang, Y., Lin, K.J.: Reputation-oriented trustworthy computing in e-commerce environments. Internet Computing, IEEE 12(4), 55 –59 (Jul-Aug 2008)
17. Wang, Y.H., Singh, M.P.: Trust Representation and Aggregation in a Distributed Agent System. In: Proceeding of National Conference on Artificial Intelligence (2006)
18. Welch, G., Bishop, G.: An introduction to the kalman filter (1995)
19. Whitby, A., Josang, A., Indulska, J.: Filtering out unfair ratings in bayesian reputation systems. In: Proceeding of the Third International Joint Conference on Autonomous Agenst Systems. pp. 106–117 (Jul 2004)
20. Xiong, L., Liu, L.: Peertrust: supporting reputation-based trust for peer-to-peer electronic communities. Knowledge and Data Engineering, IEEE Transactions on 16(7), 843 – 857 (Jul 2004)
21. Yan, S.R., Zheng, X.L., Chen, D.R., Zhang, W.Y.: User-centric trust and reputation model for personal and trusted service selection. International Journal of Intelligent Systems 26(8), 687–717 (2011)
22. Zhang, Y.C., Fang, Y.G.: A fine-grained reputation system for reliable service selection in peer-to-peer networks. Parallel and Distributed Systems, IEEE Transactions on 18(8), 1134–1145 (Aug 2007)